Math 319NT Fall 2024 **Name: Aby Soumare**
Lab 7: Dynamical Irreducibility
Collaborators:

Introduction

Within this lab, we were tasked with learning about the topic of dynamical irreducibility of polynomials over finite fields. This topic is ongoing research Professor Day and their partners are working on. This report will not contain concrete conclusions because of this fact.

Before we jump into the analysis of this lab, we first need to be familiar with some topics and questions.

First, we were introduced to a research question:

Question 0.1. *What are sufficient conditions to guarantee that a polynomial is dynamically irreducible?*

We were also tasked with creating our own problem to try and solve near the end of this lab. The question created is:

Question 0.2. *Make a list of irreducible degree 5 polynomials of the form $x^5 + x + a$ over $\boldsymbol{F}_3$. Check to see if their first, second, third, and fourth iterate are irreducible. Is it possible for any of these to be dynamically irreducible? Explain.*

You will be able to see the work done for this question at the end of this lab.

Background

Now, in order to fully grasp the concept, we will first go over terminology that you must know the definitions to, as well as understand the examples of. First off, consider:

Definition 0.3. *$\boldsymbol{F}_p[x]$ are polynomials with the coefficients of, at most, order p a prime number.*

Example 0.4. *Within $\boldsymbol{F}_5[x]$, we can generate the polynomial $x^5 + x + 1$, or even $x^5 + x^4 + x^3 + x^2 + x$.*

Now, we must also be familiar with another definition. Consider:

Definition 0.5. *An irreducible polynomial is a polynomial that cannot be factored into a polynomial of lower degree over $\boldsymbol{F}_p[x]$.*

Example 0.6. *The polynomial $x^2 - 2$ is irreducible $\in \mathbb{Z}$ because we can only reduce this down to $(x - \sqrt{2})(x + \sqrt{2})$ which is not $\in \mathbb{Z}$.*
However, we can see that this same example is reducible $\in \mathbb{R}$ since we can reduce the polynomial into lower-degree factors.

The final idea we want to understand is the topic of dynamical irreducibility. See this definition:

Definition 0.7. *Given $f \in \boldsymbol{F}_p[x]$ of degree of at least 2, we say that $f(x)$ is dynamically irreducible if*

$$f^n(x) = (f \circ f \circ f \circ .... \circ f)(x)$$
where $f \in (f \circ f \circ f \circ .... \circ f)$ repeats n times,

is irreducible for all n.

In order to proceed with the rest of the lab, make sure to understand all of these topics fully.

Analysis

First off, we got to understanding this topic with multiple exercises for us to complete. The first one being:

Exercise 0.1. *Let $f(x) = x^2 + 1 \in \boldsymbol{F}_2[x]$. Find the second iterate of $f(x) : f(f(x))$ by hand.*

So, the work for this excerise goes like this:

$$\begin{aligned} f(x) &= x^2 + 1 \in \mathbf{F}_2[x]. \\ f(f(x)) &= (x^2 + 1)^2 + 1, \\ &= (x^2 + 1)(x^2 + 1) + 1, \\ &= x^4 + 2x^2 + 2, \\ &= x^4 + 2 \end{aligned}$$

This is a simple task for very small polynomials like we were given in this example, however it would be far harder to execute if the polynomial had a higher degree. So, in order to help us compute these computations for polynomials of higher degree, we hired Magma to do the job for us. In order to make Magma do the heavy lifting for us, we used the following code to do the computation again.

Code 0.8.

```
R<x>:=PolynomialRing(FiniteField(2));
f:=x^2+1;
Evaluate(f,f);
x^4
```

Now we are to move on to another exercise that will allow us to more deeply understand the topic. See this exercise:

Exercise 0.2. *Choose your own $\boldsymbol{F}_p$ and a degree 10 polynomial. Use magma to compute the second iterate of that polynomial.*

For this, we chose the polynomial $x^{10} + 1$ and got:

Code 0.9.

```
> R<x>:=PolynomialRing(FiniteField(11));
> f:=x^10+1;
> Evaluate(f,f);
x^100 + 10*x^90 + x^80 + 10*x^70 + x^60 + 10*x^50 + x^40 + 10*x^30
+ x^20 + 10*x^10 + 2
```

Now, in order to make it so we can compute for the nth iterate, we used the code:

Code 0.10.

```
function iterate(f,n)
        if n eq 1 then
            return f;
        elif n eq 2
            return f(f);
        else
            return f(iterate(f,n-1))
        end if;
    end function;
```

We then used this code to compute if our previous polynomials were irreducible:

Exercise 0.3. *Use IsIrredcible(f) to see if:*

- *The polynomial you were using in the previous example is irreducible: $x^7 + 1$ is False*
- *The second iterate of the polynomial you used is irreducible: x^49 is False.*

Now, we are to go into the question posed at the beginning of this report.

Question 0.11. *Make a list of irreducible degree 5 polynomials of the form $x^5 + x + a$ over $\boldsymbol{F}_3$. Check to see if their first, second, third, and fourth iterate are irreducible. Is it possible for any of these to be dynamically irreducible? Explain.*

In trying to figure this question out, we first did the first part of the question, which is to find the different polynomials and check their iterates. This is what we did:

$x^5 + x + 1$	false	false	false	false
$x^5 + x + 2$	false	false	false	false
$x^5 + x + 3$	false	false	false	false
$x^5 + x + 4$	false	false	false	false
$x^5 + x + 5$	false	false	false	false

Suggesting that all of these are reducible. The reason that this is might be for the fact that none of the coefficients are going to be undefinable within

Future Directions

Since this area of study is still ongoing, then suffice it to say, there are many open-ended questions within this lab. First off, in the experiments of this lab, we only worked with polynomials of the form $x^i + x + b$, and saw that pattern that arose. However, if we were to look at polynomials of the form $x^i + ax + b$, then it would be interesting to se if the same pattern arose. Another thing, what if we were to only work with $i = p^2$ or $i = pq$ where p, q are prime numbers? Would this same result show through?

Appendix

Code 0.12.

```
> R<x>:=PolynomialRing(FiniteField(7));
>f:=x^7+1;
>Evaluate(f,f);
```

Code 0.13.

```
function iterate(f,n)
    if n eq 1 then
        return f;
    elif n eq 2
        return f(f);
    else
        return f(iterate(f,n-1))
    end if;
end function;
```